

CLAIMS

1. In a programming environment, a method for developing a transformation program to
5 transform a data structure from a first format to a second format, the program including a
plurality of coupled data transformation modules describing the transformation, the method
comprising the steps of:
generating a first transformation module of the plurality of transformation modules for
assembling the program, the first module being a module type of a set of module types including
10 a language constructed module type and a visually constructed module type;
extracting reference information from the first module for accessing the first module when stored
in a memory; and
updating a module registry to include a first entry corresponding to the reference information of
the first module, the module registry configured for having reference information entries
15 extracted from both the language constructed modules and visually constructed modules.
2. The method of claim 1 further comprising the step of storing the first module in the
memory.
- 20 3. The method of claim 2 further comprising the step of generating a second transformation
module coupled to the first module using the first entry of the module registry.
4. The method of claim 3 further comprising the step of updating the module registry to
include a second entry corresponding to reference information of the second module;
25
5. The method of claim 4, wherein the second module is of the module type different from
the first module.
6. The method of claim 5, wherein the first module is the language constructed module type
30 and the second module is the visually constructed module type.

7. The method of claim 1, wherein the module registry is a symbol table.

8. The method of claim 7 further comprising the step of including identification information in the reference information, the identification information including a name of the first reference
5 module.

9. The method of claim 8 further comprising the step of including content information in the reference information.

10 10. The method of claim 9, wherein the content information is selected from the group comprising; an argument, an argument type, and a descriptive summary of functionality of the first module.

11. The method of claim 1, wherein the first module includes a call selected from the group
15 comprising a procedure and a function.

12. The method of claim 2 further comprising the step of storing the first module in the memory in a file, the file configured for having at least two of the plurality of coupled transformation modules, the two modules being of the same module type.

20

13. In a programming environment, a system for developing a transformation program to transform a data structure from a first format to a second format, the program including a plurality of coupled data transformation modules describing the transformation, the system comprising:

25 an editor for generating a first transformation module of the plurality of transformation modules to assemble the program, the first module being a module type of a set of module types including a language constructed module type and a visually constructed module type;
a reference module for extracting reference information from the first module for accessing the first module when stored in a memory; and

a module registry for including a first entry corresponding to the reference information of the first module, the module registry configured for having reference information entries extracted from both the language constructed modules and visually constructed modules.

5 14. The system of claim 13, wherein the first module is stored in the memory.

15. The system of claim 14 further comprising a second transformation module coupled to the first module using the first entry of the module registry.

10 16. The system of claim 15, wherein the module registry includes a second entry corresponding to reference information of the second module;

17. The system of claim 16, wherein the second module is of the module type different from the first module.

15

18. The system of claim 17, wherein the first module is the language constructed module type and the second module is the visually constructed module type.

19. The system of claim 13, wherein the module registry is a symbol table.

20

20. The method of claim 19 further comprising the reference information configured to include identification information, the identification information having a name of the first reference module.

25 21. The system of claim 20 further comprising the reference information configured to include content information.

22. The system of claim 21, wherein the content information is selected from the group comprising; an argument, an argument type, and a descriptive summary of functionality of the first module.

30

23. The system of claim 13, wherein the first module includes a call selected from the group comprising a procedure and a function.

24. The system of claim 14 further comprising a file system in the memory for storing the first module in a file, the file configured for having at least two of the plurality of coupled transformation modules, the two modules being of the same module type.

25. A computer program product for developing a transformation program in a programming environment to transform a data structure from a first format to a second format, the program including a plurality of coupled data transformation modules describing the transformation, the computer program product comprising:

a computer readable medium;

an editor module stored on the medium for generating a first transformation module of the plurality of transformation modules to assemble the program, the first module being a module type of a set of module types including a language constructed module type and a visually constructed module type;

a reference module coupled to the editor module for extracting reference information from the first module for accessing the first module when stored in a memory; and

a registry module coupled to the reference module for including a first entry corresponding to the reference information of the first module, the registry module configured for having reference information entries extracted from both the language constructed modules and visually constructed modules.

26. A computer readable medium containing computer executable code for, in a programming environment, developing a transformation program to transform a data structure from a first format to a second format, the program including a plurality of coupled data transformation modules describing the transformation, the code comprising code for:

generating a first transformation module of the plurality of transformation modules for assembling the program, the first module being a module type of a set of module types including a language constructed module type and a visually constructed module type;

extracting reference information from the first module for accessing the first module when stored in a memory; and

updating a module registry to include a first entry corresponding to the reference information of the first module, the module registry configured for having reference information entries

5 extracted from both the language constructed modules and visually constructed modules.